# USER CENTRED DEVELOPMENT OF A GENERAL PRACTICE MEDICAL WORKSTATION: THE PEN&PAD EXPERIENCE

*A.L. Rector*, B. Horan*, M. Fitter†,*
*S. Kay*, P.D. Newton§, W.A. Nowlan*, D. Robinson†, and A. Wilson**

*Medical Informatics Group, Department of Computer Science,
University of Manchester, Manchester M13 9PL.
Tel: +44-61-275-6133, FAX: +44-61-275-6236, EMAIL: rector@cs.man.ac.uk
†MRC Social and Applied Psychology Unit, University of Sheffield
§Centre for Research in Primary Care, Department of General Practice, University of Manchester

**ABSTRACT**
The goal of the PEN&PAD project is to design and develop a useful and usable medical workstation for day–to–day use in patient care. The project has adopted a user centred approach and direct observations of doctors, participative design and Formative Evaluation have therefore been an integral part of the process of software development. Indeed, doctors have been involved from the earliest stages of the project. The project has focussed on British General Practitioners, but the methods which have been evolved are general. This paper describes the strategy by which doctors can be involved in the successful design and development of a medical workstation.

**KEYWORDS:** user centred, workstation, medical informatics, methodology, evaluation.

## INTRODUCTION

Doctors have often been described as resistant to using computers, and numerous surveys (for example [1]) have purported to show doctors' negative attitudes towards computers. The difficulty of producing information systems which doctors are willing to use has frequently been cited as the major barrier to the wider use of computers in medical care [2]. However, doctors are enthusiastic users of other new technology, and a recent survey in the United Kingdom suggested that although relatively few doctors use computers in the course of the medical practice, more than half of all doctors use computers at home [3].

How can this discrepancy be explained? The most obvious explanation is that doctors do find existing systems neither sufficiently useful or nor easily usable to justify the effort required to use them, whatever their feelings towards 'computers in general'. One possible solution to these inadequacies is to involve doctors more extensively in the development process. However, mere involvement is not enough, since most projects claim to have involved users, yet the problems persist.



The goal of the PEN&PAD project is to design and develop a useful and usable medical workstation for day–to–day use in patient care [4, 5]. To meet this goal, we set out to make user involvement more effective. We have developed a new methodology for involving doctors in the development process from its earliest stages. The methods are adapted from work by many others including Monk [6], Mumford [7] and Norman & Draper [8]. They have led to an iterative process of rapid prototyping and formative evaluation in which user participation is an integral part.

## THE BRITISH GENERAL PRACTICE ENVIRONMENT

Working almost entirely in primary care, the relationship between the British General Practitioner (GP) and the patient is one of 'first contact'. A GP deals with the day–to–day management of patients, tackling conditions ranging from the common cold to acute diabetes. The GP also has the role of referring patients to other hospital–based consultants. A typical British General Practice will consist of 3 full–time doctors, a nurse, a practice manager, and auxiliary administrative staff.

## CIRCLES OF USERS AND THE ITERATIVE DEVELOPMENT CYCLE

The user centred strategy employed by the PEN&PAD project relies on the long–term involvement of General Practitioners (GPs). The GPs are described in terms of four categories or 'circles' of doctors; the iterative process embodied in this strategy is described in terms of 'cycles'.

### The General Practice Users

There are four categories of GP users:

i) One doctor is employed full time as a member of the development team[1].

ii) An 'inner circle' of five doctors provides the primary input to the design and evaluation process. They are involved in workshops which take place roughly once per month and in individual sessions and preparation equivalent to two or three additional days per month. The doctors

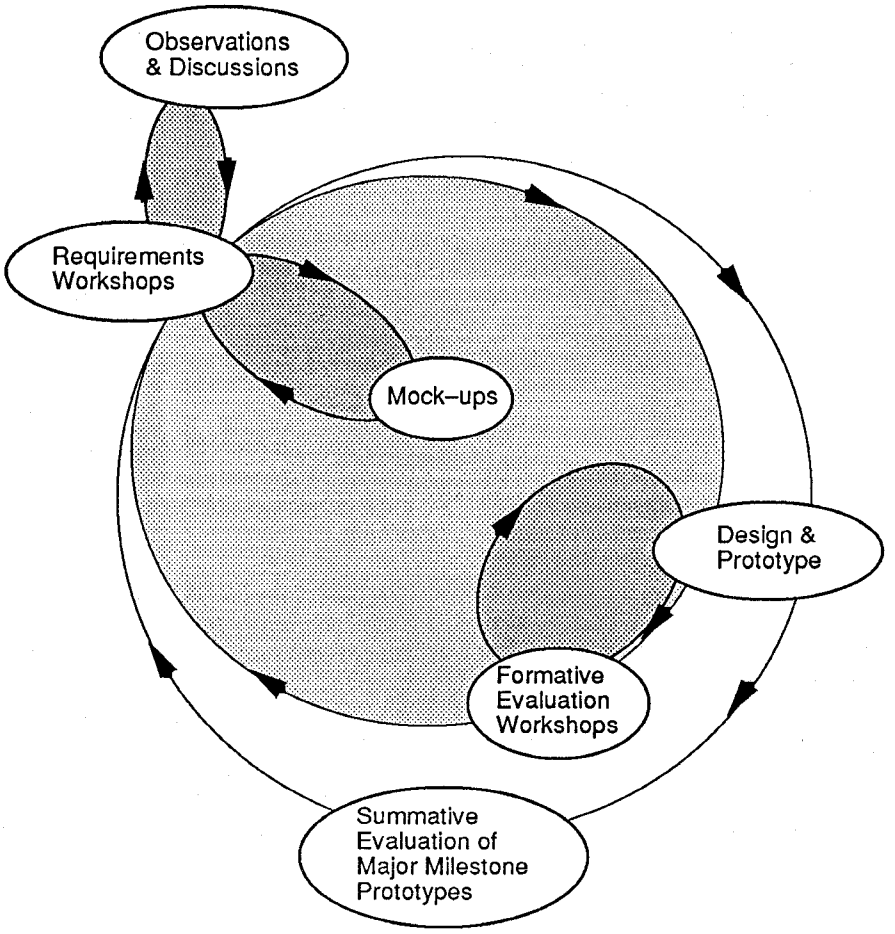


**Figure 1: A schematic representation of the User Centred Development Process**

were selected to include a range of types of general practice and of attitudes towards computers. All of them have, or have acquired during the course of the study, computers in their practices for at leastadministrative purposes[2]. None of them is involved in academic practice.

iii) An 'outer circle' of twelve doctors participates in major formative evaluation workshops which are held at roughly six–monthly intervals. These doctors were drawn from the British Computer Society Primary Health Care Specialist Group, the primary organisation of general practitioners interested in computing in the UK, augmented by members of the department of General Practice at the University of Manchester.

iv) A group of eight 'naive' users, doctors with no prior connection with the project, who participate in the final 'quasi–summative' phase of the evaluation. These doctors, whose characteristics are similar to the group above, were unpaid. Three had no computer experience whatsoever, the remainder used computers in their consulting room.

**The Formative Evaluation Cycle**

The project makes a distinction between 'formative' and 'summative' evaluation [9]. 'Formative evaluation' is undertaken for the development team and aims to refine the requirements and thus improve the design. 'Summative evaluation' is carried out either for the development team or for external agencies to determine whether or not the project has reached pre–defined criteria. Formative evaluation is an integral part of the development cycle and is also used to refine the methods employed in the summative evaluation.

The development strategy is shown schematically in figure 1. Each cycle consists of a series of workshops for requirements analysis and preliminary design specification, then a short period in which software is produced, followed by a formative evaluation workshop.

In the first cycle of the development process, a series of workshops was held to determine initial user requirements and to outline proposed designs. The workshops consisted of group discussions supplemented by informal demonstrations and assessments of new ideas. A minority of the demonstrations were computer–based; most relied on paper or overhead transparencies. The design exercises were focussed on doc-

tors' tasks, and their requirements for medical information. In addition, all members of the development team spent time observing doctor–patient consultations.

Based on this information, the development team produced initial computer–based prototypes. The prototypes were largely 'cardboard'—little more than slide shows to demonstrate potential. They were shown first to the inner circle to eliminate the worst flaws in the user interface, and then subjected to a formative evaluation workshop with the outer circle of doctors.

After the formative evaluation workshop, the results were analysed and fed into the next cycle of the development process. A further series of requirements analysis and design workshops was held to explore issues raised by the evaluation workshop and refine the development team's understanding of the requirements. In addition, the workshops were used to propose increasing degrees of functionality for the next prototype. The current prototype was then completely discarded and the new one was implemented from scratch during the next cycle.

At the end of the third cycle, it was felt that the system had reached a plateau in its development. It was also feared that the doctors in both the inner and outer circles were becoming too involved in the project to give unbiased advice. The approach described in this paper stresses the need to involve users in the development process and to thus produce a 'shared culture' between themselves and the developers. One advantage of this is a common language that can enhance the understanding of some aspects of the problem. However, it can also create a set of 'blinkers' which prevent both parties from seeing certain blunders or faults [9]. A series of 'quasi–summative' evaluation workshops were therefore held which involved doctors who had no previous contact with the system and who were chosen to be more representative of the wider GP community. We termed this evaluation 'quasi–summative' because the system was far from complete.

## THE WORKSHOP METHODOLOGY

The workshops involve three groups:

i) The development team who produce the prototypes;

ii) The evaluation team consisting of psychologists and associated observers;

iii) The doctors.

The evaluation team is separate from the development team, but participates in many of the project planning activities. They are, in general, viewed by the development team as colleagues and allies, although they also have a separate responsibility to prepare a final evaluation report for the funding bodies.

### The Requirements Analysis and Design Workshops

The pattern of the Requirements Analysis and Design Workshops is highly variable. Each workshop is devoted to a particular design issue. On most occasions, the issue relates to a prototype which the doctors have seen at a previous workshop. On other occasions, the issue considered has been raised at a previous workshop or during observations and discussions with the doctors outside the formal workshops. Examples of the specific issues discussed are:

- the flow of the consultation;
- what data needs to be entered at each stage of the consultation;
- the degree to which the information presented needs to depend on the context in which it is to be used.

Each doctor is expected to spend at least one paid session before each workshop preparing material and ideas. Frequently, the doctors find it convenient to demonstrate their ideas in the form of screen scenarios created on overhead transparencies. On other occasions, particularly when they are working on the details of the information required in particular areas, they find it more helpful to prepare handouts to be circulated to their colleagues prior to the workshop.

The requirements analysis and design workshops are led by the medical member of the development team and attended by the inner circle doctors, the development team, and the evaluation team. Each workshop consists of a two hour discussion in which the doctors discuss the issue in question with the occasional participation of members of the development and evaluation teams. In many sessions, the development team spend a significant amount of time checking their conclusions with the doctors. The normal outcome of a workshop is a commitment from the development team to prepare additional material or prototypes. Frequently, there is also a commitment from the doctors to prepare more material for a subsequent workshop.

### The Formative Evaluation Workshops

The formative evaluation workshops follow a much more standardised format. They are led by the evaluation team and involve doctors from both the inner and outer circles. Some of the workshops also include one or two 'naive' doctors who have not seen the system before. A fundamental principle of the formative evaluation workshops is that the workshops are 'owned' by the evaluators and the users, not the developers. At each workshop, the schedule of events is similar. After a brief introduction, the new prototype is demonstrated by a member of the development team. The doctors are then divided into pairs, and a member of the development team and an observer from the evaluation team are assigned to each pair of doctors. Each pair of doctors is provided with specific 'training tasks' designed to familiarize them with the features of the system. Some doctors choose to perform the tasks quite literally, others use them merely as a starting point for a more general exploration.

The doctors are then given specific 'evaluation tasks' to perform. In almost all cases, these tasks are in the form of role–playing exercises in which one user takes the role of "patient" and the other takes the role of "doctor". The "doctor" is asked to use the system in the course of performing a

consultation with the "patient". The development team and evaluation team cooperate to prepare scenarios in advance. Experience shows that most doctors are able to improvise the role of patient from very brief vignettes and that the variation which the doctors' different interpretations introduce into the evaluation task provides valuable extra information on how the system might be used in alternative situations.

Following the evaluation tasks, the doctors complete a short questionnaire and structured interview schedule. There is usually, either spontaneously or formally, a brief discussion between the two doctors and the member of the evaluation team. Throughout the training and evaluation tasks, the members of the development team are restricted to providing only 'on line help' in order to let the doctors complete their tasks. Members of the development team are not allowed to discuss the reasons for different features of the design and are not, above all, to argue about whether or not particular features are sensible, easy to use, consistent, or convenient.

Following the completion of the questionnaires, the doctors, development and evaluation teams meet in a group discussion conducted by the leader of the evaluation team. The first part of this discussion is devoted to high level issues concerning the prototype. After the initial discussion, the development team are officially asked to join in, and much of the subsequent discussion consists of a dialogue between the development team and the users concerning the reasoning behind various features. Finally, time permitting, there is a brief 'brain–storming' session to capture any new ideas which might have been stimulated by the day's activities.

Following the workshop, there is a debriefing session for the development and evaluation teams to gather the immediate responses and impressions. Over the next few days, their observations are written up by all members of the development team, and the evaluation team prepare a formal report for the development team. During the subsequent design and evaluation meetings, these results are used to modify the task analysis, user requirements, and design.

## RESULTS OF THE USER CENTRED DEVELOPMENT PROCESS

### Education of the Development Team: Formative Evaluation as Process not Product

The most important result of the user centred development process was that the development team came to understand the doctors' problems. This effect is difficult to quantify, but there is unanimous agreement amongst members of the development team that the workshop process has been instrumental in overcoming their misconceptions concerning medical practice and doctors' needs. The workshops are dramatically more effective in achieving this than mere demonstrations, in which the same team frequently take part. Watching users attempt to achieve realistic tasks with the system is a very powerful motivator to remove problems.

The most dramatic example of the development team learning about the users' needs has been the acceptance of the extent of user variability. Developers inevitably look for the 'best' way to complete a task and become attached to their pet notions. Invariably we have found that some situations, or some users, require an alternative.

The workshops are also effective in keeping the development team focussed on the doctors' priorities—'keeping their feet on the ground'. All development projects have a natural tendency to create their own agendas. Each workshop provides a valuable corrective.

An important concomitant result is that the inner circle of doctors has come to understand the development team's thinking and to understand the prototypes in much more detail than would ordinary users. They are frequently able to offer valuable suggestions which would be impossible for more naive users.

#### Specific Results

The evaluation workshop produces three types of output:

- 'Micro level' diagnostic comments on particular ergonomic details which need to be fixed. These comments describe the usability of a particular feature with little regard for its context.
- 'Midi level' comments which tend to be comparative. These relate the current implementation of some large subsystem (such as data entry or chart handling) to its previous implementation.
- 'Macro level' comments which assess the impact of the system on the wider medical environment, the likely response of colleagues, patients, and staff, and the likely impact on the quality of care, organisation of the surgery, and relations with mangers and hospital consultants.

Most workshops uncover a number of micro level problems. Where the development team believe that a micro level defect is so severe that the evaluation of the higher levels is invalid, a subsequent informal assessment is held after the fault is fixed, so that the overall evaluation does not have to await the next major workshop[3].

The workshop tasks are designed primarily to elicit midi level comments: these involve the use of the system both to perform these tasks and to navigate within and between them. The outcome of these comments is a further refinement by the development team of its analysis of the task structure of the consultation. Table 1 gives details of some solutions to the micro and midi level problems raised by previous workshops. The structured interviews and subsequent workshop discussion give rise to macro level comments. Overall, these have produced three major results:

- Most doctors liked the systems and say that we seem to be 'on the right track'.

| Micro | Midi |
|---|---|
| Provide 'clear up' button to remove all but master window | Transform 'scratch pad' into 'comment' area on data entry forms |
| Automatically position collapsed windows | Indicate normal range on the display of numeric data |
| Improve switching between graphical and tabular charts | Include the number of available results for each item on the menu of results |
| Employ users' language, e.g. 'tds' instead of 'three per day' | Adopt model of over–inclusiveness in determining content of data entry forms |

**Table 1: Some changes at the micro and midi level resulting from the formative evaluation workshops.**

- There is no one best way. Medical practice is enormously complicated, and users are highly variable. Consequently, systems must be equally flexible.

- Simple clear presentations are often more effective than sophisticated attempts to provide intelligent summaries. Doctors are remarkably good at recognising patterns if the information is clearly presented.

For the development team the most important outcome of the user centred process is the second of these macro level results: the realisation of the range of variability of users and of medical situations. For example, the user centred process made it obvious that some doctors find the use of a mouse difficult, some are well accustomed to keyboards and good typists, and in some situations, typing a short abbreviation is faster than any conceivable search through menus or forms. Without the user centred process, the project would almost certainly have followed its original remit and produced a system which could only be used with a mouse. A prime requirement now is that all functions be equally accessible from the keyboard.

In addition to the macro level results, the development and evaluation team abstracted a series of overall constraints or 'maxims' for the development of the system from the micro and midi level comments. These maxims reflect our understanding of the basic properties which the system must possess to be usable in medical practice. They include items such as:

- The system must always be interruptable. This means that if the user moves his or her hands completely away from the system, it must be in precisely the same state when he or she returns to it. Therefore, for example, all menus pop up and stay up even when the mouse button is released.

- All, or almost all, options should be immediately visible. The use of multi–state buttons and similar 'hidden' options should be minimized.

- Everything on the screen should convey information about the patient being seen. The doctor should never ask for something—e.g. the result of a test—only to discover that it does not exist in the medical record.

- In organising information and forms, it is better to be inclusive than exclusive. Within limits, doctors will happily ignore superfluous items, provided they do not have to respond to them in any way. (In this way a pointer–based interface which allows free 'random access' to the screen produces very different results from a keyboard based interface in which, for example, a user must use the cursor keys to skip irrelevant items.)

- There should be no 'modes'. All functions should be possible at every stage of the consultation, since the course of the consultation is highly variable and unpredictable.

### Interaction with Project Management

One of the most important effects of the iterative user centred process is that the Formative Evaluation Workshops provide absolute deadlines and milestones for the project. The workshops have to be scheduled several months in advance in order to fit into the participants' diaries. Once scheduled, they are virtually immovable. The goals of the prototype might be scaled down, but a working prototype has to be completed by the given date. The Formative Evaluation Workshops are highly public events and provide much harder deadlines than any internal project management process.

## PROBLEMS AND LIMITATIONS

### Problem Space vs Solution Space: or "Users are always right... except when they're wrong"

The most serious problem for anyone attempting to involve users actively in software development is how to treat the multitude of requests and ideas which emerge. Users' groups are notorious for generating endless lists of requests, or 'wish–lists', which suppliers cannot hope to meet.

The PEN&PAD approach to this problem is to divide the conceptual space into

- The problem space which is the province of the user. The problem space is concerned with the needs of the users in terms of their *requirements*, the variability that exists in those requirements, and the usability or acceptability of the system, which can only be discovered by evaluating the *use* of the system in a semi–realistic context.

- The solution or design space which is the province of the development team. The solution space is concerned with the creative design *ideas* generated by the developers and the constraints of practicality and feasibility implicit in the *technology* in which those ideas are implemented.

Users are always right about the problem space. If users say they cannot understand something or that they need to do something—e.g. that they do not understand the meaning of command buttons or if they say that they need a 'scratch pad'—then they must be believed[4].

However, users are usually wrong in their suggestions as to how best to meet their needs or remedy a problem. Users' comments are usually concrete and framed in terms of the problem space, whereas a good design solution is likely to be abstract and must be framed in terms of the design space.

The example of the users' request for a 'scratch pad' is illuminating.

The request for a 'scratch pad' was made early in the project and resulted in the addition of a large free–text window to the second prototype. However, in the formative evaluation exercises, the doctors made almost no use of this facility. Furthermore, it was obvious from the doctors' comments that the literal 'scratch pad' did not serve the need that had provoked the request.

We took the topic of the 'scratch pad' as the subject of one of the Requirements Analysis and Design workshops. It rapidly emerged that there were at least three different functions needed:

- A diary or agenda on which the doctor could record reminders—e.g. to telephone a consultant or make arrangements with social services following the surgery.
- A place for private notes which were not part of the patient's official medical record but of which the doctor wished to be reminded, e.g. 'ask about his cat' or 'Remember to thank her for the Christmas card.' The doctors also wanted to use this space for reminders of things which were worrying them but which they did not want to commit to the medical record because they were too tentative.
- A place for completely private, protected, and confidential items which should not appear even in the most secure part of the patient's medical record.

This example of the scratch pad is typical. Repeatedly, users' requests which were superficially simple, turned out to hide a multiplicity of functional requirements. Conversely, it was often possible to meet a large number of requests with a single design feature. The most dramatic example of a single feature eliminating many individual requests was the provision of a 'clear up' button to remove all the windows except that containing the patient summary. Providing the 'clear up' button eliminated the need for most of the other, more complex, features which had been requested for controlling layout and 'clutter'.

### Difficulties with the Prototyping Process

The use of prototypes gives rise to a number of intrinsic difficulties. Users and outside observers have difficulty determining which parts of a prototype are 'real' and which are simply well–crafted mock–ups. This frequently leads the users to have unrealistic expectations as to how rapidly a working system can be produced.

More significantly, users find it difficult to distinguish between those features of a prototype which the development team expects to carry through to later versions and those which are merely accidents of the particular prototyping tools being used. The result can be that the users spend much time discussing things of no relevance to the further design of the system.

The most extreme example of this problem occurred at the first Formative Evaluation Workshop in which the logging–on procedure was particularly unintuitive to users not already familiar with the conventions of the development environment. The development team had been aware that this procedure was awkward, but had not thought it important. However, nearly half the discussion time was taken up with comments relating to logging–on to the system and to subsequent difficulties which followed from it. A similar example is the difficulty which many users experience with an optical mouse[5]. Users who have this difficulty must be quickly moved to a machine with a mechanical mouse or their comments will be almost worthless.

### The Role of Medical Content

In a similar way, the medical content of the scenarios and the prototypes may cause controversies which obscure the issues that the evaluation and development teams wish to explore. Doctors are naturally more concerned about medical issues than about computing science and arguments about medicine can easily come to dominate the discussion. For example, a workshop intended to explore various alternative presentations of clinical protocols can easily be side–tracked by discussion of the controversy concerning how aggressively moderate high blood pressure should be treated. Wherever possible we have tried to use example scenarios which minimize this type of medical controversy and focus on the use of the system itself.

### Lack of Experienced Users

The biggest defect in the user centred strategy is that none of the users ever becomes really expert in using the system. There is an enormous difference between using the system for a few hours during a formative evaluation workshop and using it routinely, several hours per day, five days per week.

## DISCUSSION

To date, the medical informatics community has not been notably successful in producing systems which are widely used in routine medical practice. It is all to easy to blame the doctors for these difficulties, adopting an attitude which might be caricatured as 'Pearls before swine'. The alternative explanation for this lack of success is that our systems have rarely actually met medical requirements or been usable in clinical conditions—an attitude which might be caricatured as 'The emperor's new clothes". The second attitude is far more constructive for members of the informatics community, because it leads to important consequences for the development process.

The User Centred Development and Formative Evaluation methodology reported here has been created in an attempt to take the "emperor's new clothes" attitude seriously. PEN&PAD has not yet been used in real medical settings, so we cannot guarantee that the result is a practical system which doctors will use. However, we can say that the User Centred Development process has helped us to avoid a number of serious blunders which would have rendered the eventual system unusable and that the system has been enthusiastically received in all of our simulations and demonstrations to date. The next step is to put more advanced prototypes into the field and observe how they are used.

The Formative Evaluation Workshops have also been used as part of two European collaborative projects, EURODIABETA and PRECISE. In both cases, they were used in isolation to assess systems which were already well advanced in their construction. These experiences confirmed the effectiveness of the simulations in eliciting convincing responses from doctors, but they revealed significant problems in extending the methodology to other sites where it is not part of the culture. Within the project, the development team and evaluation team have negotiated their relationship over the life of the project, and the evaluation team are familiar with the goals and needs of the development team. No such process had taken place with the other projects evaluated during the European projects, and significant conflict arose concerning the remit of the formative evaluation and the goals of the systems being evaluated. Nonetheless, most participants agreed that the process had been helpful and had resulted in important new insights.

What is the key ingredient in the process? The developers are often asked if they could not elicit the same information from simple demonstrations, and they invariably reply emphatically that they could not. The atmosphere and information gathered in the two situations is entirely different. We believe the key ingredient is the formal structure provided by the workshops and the presence of external evaluators. The demonstrations are 'owned' by the development team but the workshops are 'owned' by the users. The fact that workshops are conducted by the evaluators rather than the developers; the fact that the users must actually attempt to use the system for a practical task; and the rule that the developers can only act as 'on line help' combine to create an atmosphere in which users provide much more frank and honest criticism than in demonstrations or less formal settings.

## ACKNOWLEDGEMENTS

This research supported in part by grants in the United Kingdom from the Medical Research Council number SPG8800091, from the Department of Health, and from the European Community initiative on Advanced Informatics in Medicine (AIM) as part of the EURODIABETA and PRECISE projects.

## FOOTNOTES

[1] The development team are responsible for the design of the clinical and its implementation.

[2] By 1991 approximately 50% of all British general practitioners had a practice computer system according to UK Department of Health sources.

[3] Note that micro level comments are almost always negative. Good micro–ergonomic design is invisible. The team discovered this to its cost when in the first version of the latest prototype, a number of 'minor' ergonomic features were changed, producing a system which was universally disliked by both our regular and 'naive' users until the original micro–ergonomics features could be restored.

[4] As obvious as it seems to say that there is no point in arguing with somebody about whether or not they understand something or find it easy, we have frequently observed developers doing precisely that when users failed to appreciate one of their pet features. Despite the rules that developers were only to act as 'on line help' during the exercises, there were times when they could hardly restrain themselves. Observations of other groups and demonstrations suggest that our group is far from unique. It was our experience with such discussions early in the development of the Formative Evaluation Workshops which led us to institute the rule that developers could only observe during the formal evaluation sessions.

[5] Without exception, all clinical users expressed a preference for the Macintosh mechanical mouse over the Sun optical mouse. Although most users were able to use both, there was a minority of users who found the optical mouse completely unusable.

## REFERENCES

1. Teach, R.L., Shortliffe E.H. An analysis of physician attitudes regarding computer based clinical consultation systems. Comp Biomed Res, 1981;14, 542–58.
2. Shortliffe, E.H. Computer Programs to Support Clinical Decision Making. Journal of the American Medical Association, 1987;258, 63–68.
3. Young, D., Chapman, T., and Poile, C. Physician, reveal thyself. British Journal of Healthcare Computing, 1990,7,9,16–21.
4. Howkins, T.J., Kay, S., Goble, C.A. et al. An Overview of the PEN and PAD Projects. In Moore, R.O., Bengtsson, S., Bryant, J.R., and Bryden, J.S., eds. Lecture Notes in Medical Informatics 40, Medical Informatics Europe 90, Glasgow. Berlin, Springer–Verlag, 1990,73–78.
5. Horan, B., Rector, A.L., Sneath, E.L. et al. Supporting a Humanly Impossible Task: The Clinical Human Computer Environment. In Diaper, D., Gilmore, D., Cockton, G., and Shackel, B., eds. Human Computer Interaction—Proceedings of INTERACT'90. Amsterdam, Elsevier Science, 1990,247–252.
6. Monk, A. Fundamentals of Human Computer Interaction. London, Academic Press, 1984.
7. Mumford, E. Designing Human Systems for New Technology: The ETHICS Method. Manchester, Manchester Business School Press, 1983.
8. Norman, D.A., and Draper, S.W. User Centred System Design: New Perspectives on Human Computer Interaction. Hillsdale, New Jersey, Lawrence Erlbaum Associates, 1986.
9. Hewett, T.T. The Role of Iterative Evaluation in Designing Systems for Usability. In Harrison, M.D. and Monk, A.F., eds. People and Computers II: Designing for Usability. Cambridge, Cambridge University Press, 1986, 196–214.